Data Analysis Higher Education Wages

his dataset contains salary information for employees of the Pennsylvania State System of Higher Education in 2013

Name: The name of the employee. (String)
Base Pay: The base salary of the employee. (Numeric)
University or Office: The university or office the employee works for. (String)
Position: The position of the employee. (String)

https://www.kaggle.com/datasets/thedevastator/uncovering-wage-disparities-in-pennsylvania-s-hi/data

```
df
```

	index	Name	Base Pay	University or Office	Position
0	0	Driscoll, Michael Allan	$275000	Indiana	University President
1	1	Weisenstein, Greg R	$241935	West Chester	University President
2	2	Welsh, Marcia L	$230000	East Stroudsburg	University President
3	3	Norton, Cheryl J.	$225000	Slippery Rock	University President
4	4	Wollman, Julie E.	$225000	Edinboro	University President
...	...	...	...	...	...
12627	12627	Black, Jacqueline D	$1254	Edinboro	Instructor, PT Temp
12628	12628	Webster, Kim	$1245	Kutztown	Instructor
12629	12629	Hanelly, William T	$152950	Lock Haven	Vice President/Fin.Admin
12630	12630	Moran, James D Iii	$193125	Office of the Chancellor	Vice Chancellor Academic & Student Affairs
12631	12631	Cavanaugh, John C	$327500	Office of the Chancellor	Chancellor

12632 rows × 5 columns

For convenience, we will change the names of the columns.For convenience, we will change the names of the columns, and also check the format of the columns and replace float with int if necessary

```
[378] df.columns = df.columns.str.replace (' ','_').str.lower()
```

```
df
```

	index	name	base_pay	university_or_office	position
0	0	Driscoll, Michael Allan	$275000	Indiana	University President
1	1	Weisenstein, Greg R	$241935	West Chester	University President
2	2	Welsh, Marcia L	$230000	East Stroudsburg	University President
3	3	Norton, Cheryl J.	$225000	Slippery Rock	University President
4	4	Wollman, Julie E.	$225000	Edinboro	University President
...	...	...	...	...	...
12627	12627	Black, Jacqueline D	$1254	Edinboro	Instructor, PT Temp
12628	12628	Webster, Kim	$1245	Kutztown	Instructor
12629	12629	Hanelly, William T	$152950	Lock Haven	Vice President/Fin.Admin
12630	12630	Moran, James D Iii	$193125	Office of the Chancellor	Vice Chancellor Academic & Student Affairs
12631	12631	Cavanaugh, John C	$327500	Office of the Chancellor	Chancellor

12632 rows × 5 columns

Next, let's check the format of the strings

```
[380] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12632 entries, 0 to 12631
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   index                 12632 non-null  int64
 1   name                  12632 non-null  object
 2   base_pay              12632 non-null  object
 3   university_or_office  12632 non-null  object
 4   position              12632 non-null  object
dtypes: int64(1), object(4)
memory usage: 493.6+ KB
```

We need to change the format of the salary line and convert it to numeric.

But first, let's clean up the dollar symbol

```
[382] df['base_pay'] = df['base_pay'].str.strip('$')
```

```
df
```

	index	name	base_pay	university_or_office	position
0	0	Driscoll, Michael Allan	275000	Indiana	University President
1	1	Weisenstein, Greg R	241935	West Chester	University President
2	2	Welsh, Marcia L	230000	East Stroudsburg	University President
3	3	Norton, Cheryl J.	225000	Slippery Rock	University President
4	4	Wollman, Julie E.	225000	Edinboro	University President
...	...	...	...	...	...
12627	12627	Black, Jacqueline D	1254	Edinboro	Instructor, PT Temp
12628	12628	Webster, Kim	1245	Kutztown	Instructor
12629	12629	Hanelly, William T	152950	Lock Haven	Vice President/Fin.Admin
12630	12630	Moran, James D Iii	193125	Office of the Chancellor	Vice Chancellor Academic & Student Affairs
12631	12631	Cavanaugh, John C	327500	Office of the Chancellor	Chancellor

12632 rows × 5 columns

```
[384] df['base_pay'] = df['base_pay'].astype(str).astype(int)
```

Let's look at the salary distribution

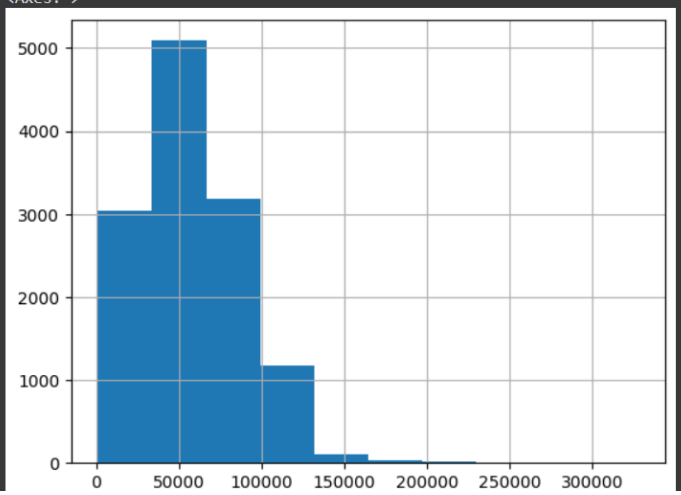
The graph shows that the distribution does not obey normal, so it is better to consider the median

```
[385] df['base_pay'].agg(['mean','median'])
```

```
mean      57575.800823
median    51856.000000
Name: base_pay, dtype: float64
```

```
df['base_pay'].hist()
```

```
<Axes: >
```



Let's see what salaries are in universities and offices and draw a conclusion

```python
df.groupby('university_or_office')['base_pay'].agg('median').sort_values( ascending=True)
```

```
university_or_office
Mansfield                    47035.0
West Chester                 47142.5
Millersville                 47297.5
California                   48343.5
Cheyney                      48489.0
Bloomsburg                   49387.0
Edinboro                     50660.0
Clarion                      51736.0
Indiana                      51856.0
Kutztown                     52249.5
East Stroudsburg             52390.0
Shippensburg                 54000.0
Slippery Rock                55810.0
Lock Haven                   56245.0
Office of the Chancellor     73228.0
Name: base_pay, dtype: float64
```

The graph shows that the highest salaries in Office of the Chancellor

```python
df.groupby('university_or_office')['base_pay'].median().plot(rot=100,ylim=0,grid=True, kind='bar')
```

```
<Axes: xlabel='university_or_office'>
```